USB Bus Interface Chip CH375

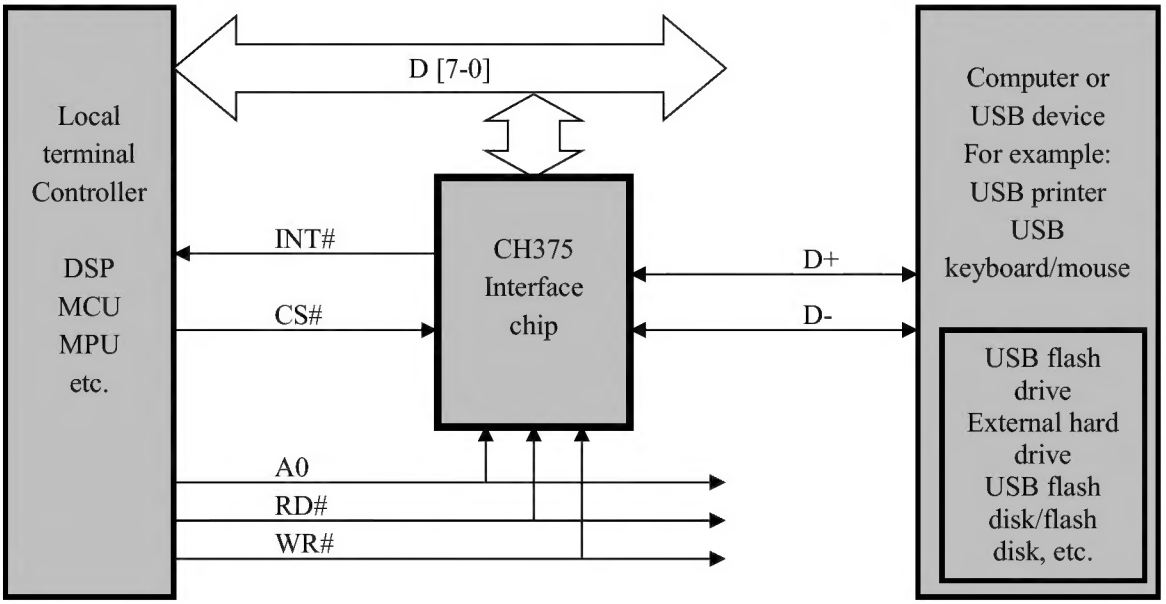
Datasheet
Version: 4
http://wch.cn

1. Overview

CH375 is a USB bus universal interface chip, which supports USB-HOST mode and USB-Device /SLAVE mode. At the local end, CH375 has an8-bit data bus, a read, write and chip selection control line and interrupt output, and can be easily hooked to the system bus of DSP/MCU/MPU and other controllers. In USB host mode, CH375 also provides the serial communication mode, being connected with DSP/MCU/MPU through serial input, serial output and interrupt output.

The USB device mode of CH375 is fully compatible with CH372, and CH375 includes all functions of CH372. This datasheet does not provide the description of CH375 in USB device mode. For related information, refer to CH372 Datasheet CH372DS1.PDF.

The USB host mode of CH375 supports commonly used USB full-speed devices, and the external MCU can communicate with USB devices through CH375 according to the corresponding USB protocol. CH375 also has the internal firmware for processing the special communication protocol of Mass-Storage device. The external MCU can read and write the commonly used USB storage devices (including USB hard disk /USB flash drive/USB flash disk) directly in sectors.



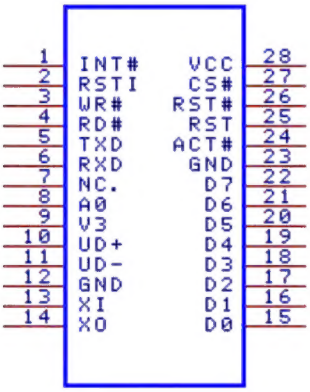
2. Features

- Low-speed and full-speed USB-HOST interface, compatible with USB V2.0; only crystal and capacitors are required for peripheral components.
- Low-speed and full-speed USB device interface, fully compatible with CH372, supporting dynamic switching between host mode and device mode.
- The host endpoint input and output buffers are respectively 64 bytes, supporting 12Mbps full-speed USB devices and 1.5Mbps low-speed devices.
- Support USB device control transmission, bulk transmission and interrupt transmission.
- Automatically detecting the connection and disconnection of USB device, and providing event

notifications for device connection and disconnection.

- Built-in control transmission protocol processor simplifies common control transmission.
- The built-in firmware processes special communication protocols for mass storage devices, and supports BULk-Only transport protocol and USB storage devices (including USB hard disk/USB flash drive/USB flash disk/USB card reader) with SCSI, UFI, RBC or equivalent command sets.
- MCU reads and writes files in the USB storage device through the USB flash disk file level subroutine library.
- The parallel interface contains an 8-bit data bus and a 4-wire control: read strobe, write strobe, chip selection input and interrupt output.
- The serial interface includes serial input, serial output and interrupt output, and supports the dynamic adjustment of communication baud rate.
- Support the supply voltage of 5V or 3.3V, and support the low-power mode.
- Adopt SOP-28 lead-free package, compatible with RoHS, provide SOP28 to DIP28 conversion board, and pins are basically compatible with CH374 chip.

3. Package



Package	Width of Plastic		Pitch of Pin		Instruction of Package	Ordering Information
SOP-28	7.62mm	300mil	1.27mm	50mil	Standard 28-pin patch	CH375C
SOP-28	7.62mm	300mil	1.27mm	50mil	Standard 28-pin patch	CH375B

Difference: CH375B must be externally connected with a crystal and an oscillating capacitor;

CH375C can be externally connected to a crystal and a capacitor, or use the built-in clock directly without external connection.

4. Pins

Pin No.	Pin Name	Pin Type	Description
28	VCC	Power	Positive power input, an external 0.1uF power decoupling capacitor is required
12, 23	GND	Power	Common ground, shall be connected to the ground wire of the USB bus
9	V3	Power	Connected to the VCC input external power at the supply voltage of 3.3V Connected to an external decoupling capacitor with capacity of 0.01uF～0.1uF at 5V supply voltage

13	XI	Input	Input of the crystal oscillator, required to be connected to an external crystal and oscillation capacitor. In the internal clock mode of CH375C, XI shall be connected to GND
14	XO	Output	Reversed output of the crystal oscillator, required to be connected to an external crystal and oscillation capacitor. In the internal clock mode of CH375C, XO shall be suspended
10	UD+	USB signal	USB bus D+data line
11	UD-	USB signal	USB bus D - data cable
22～15	D7～D0	Bi-directional Tri-state	8-bit bidirectional data bus, with built-in weak pull-up resistor,
4	RD#	Input	Read strobe input, active low, with built-in weak pull-up resistor
3	WR#	Input	Write strobe input, active low, with built-in weak pull-up resistor
27	CS#	Input	Chip selection control input, active low, with built-in weak pull-up resistor
1	INT#	Output	Interrupt request output after reset, active low
8	A0	Input	Address line input, distinctive command port and data port, with built-in weak pull-up resistor, Write commands when A0=1; write/read data when A0=0
24	ACT#	Output	USB device configuration completion status output in USB device mode with built-in firmware, active low. USB device connection status output in USB host mode, active low
5	TXD	Input Output	It is only used in USB host mode, and only parallel port supports in device mode, Input pin during reset, with built-in weak pull-up resistor, If the low level is input during reset, enable the parallel port. Otherwise, enable the serial port. Serial data output after reset
6	RXD	Input	Serial data input, with built-in weak pull-up resistor
2	RSTI	Input	External reset input, active high, with built-in pull-down resistor
25	RST	Output	Power-on rest output and external reset output, active high
26	RST#	Output	Power-on rest output and external reset output, active low
7	NC.	Idle pin	Idle pin, must be suspended

5. Commands

For the data in this datasheet, those with suffix B are binary number, and those with suffix H are hexadecimal number. Otherwise, it is a decimal number.

MCU in this datasheet is basically applicable to DSP or MCU/MPU/SCM, etc.

This datasheet mainly provides USB storage device commands, which are used exclusively for USB mass

storage devices. Please refer to Datasheet (II) for execution of transaction commands and commonly used control transmission commands. Please refer to CH372 Datasheet for USB device mode commands.

Code	Command Name	Input Data	Output Data	Command Purpose
01H	GET_IC_VER		Version	Get the chip and firmware versions
02H	SET_BAUDRATE	Frequency division coefficient Frequency division constant	(Wait for 1mS) Operation status	Set serial communication baud rate
03H	ENTER_SLEEP			Enter low-power sleep suspend state
05H	RESET_ALL		(Wait for 40mS)	Execute hardware reset
06H	CHECK_EXIST	Any data	Bitwise NOT	Test working status
0AH	GET_MAX_LUN	Data 38H	Maximum unit number	Get the maximum logical unit number of the USB storage device
0BH	SET_DISK_LUN	Data 34H Logical unit number		Set the current logical unit number of the USB storage device
0BH	SET_PKT_P_SEC	Data 39H Number of packets in each sector		Set the current logical unit number Total number of data packets in each sector
15H	SET_USB_MODE	Mode code	(Wait for 20uS) Operation status	Set USB working mode
16H	TEST_CONNECT		(Wait for 2uS) Connection status	Check the connection status of USB device
17H	ABORT_NAK			Abort the retry of the current NAK.
22H	GET_STATUS		Interrupt status	Get the interrupt status and cancel the request
28H	RD_USB_DATA		Data Length Data stream	Read the data block from the endpoint buffer of the current USB interrupt
2BH	WR_USB_DATA7	Data Length Data stream		Write data blocks to the output buffer of the USB host endpoint
51H	DISK_INIT		Generate interrupt	Initialize USB storage device
53H	DISK_SIZE		Generate interrupt	Get the capacity of the USB storage device
54H	DISK_READ	LBA address Number of sectors	Generate interrupt	Read data blocks from the USB storage device
55H	DISK_RD_GO		Generate interrupt	Continue the read operation of the USB storage device

56H	DISK_WRITE	LBA address Number of sectors	Generate interrupt	Write data blocks to the USB storage device
57H	DISK_WR_GO		Generate interrupt	Continue the write operation of the USB storage device
58H	DISK_INQUIRY		Generate interrupt	Inquire the features of USB storage device
59H	DISK_READY		Generate interrupt	Check whether the USB storage device is ready
5AH	DISK_R_SENSE		Generate interrupt	Check the error of the USB storage device

If the output data of the command is the operation status, refer to the following table.

Status code	Status name	Status description
51H	CMD_RET_SUCCESS	Operated Successfully
5FH	CMD_RET_ABORT	Operation failure

5.1. Command GET_IC_VER

This command is used to get the chip and firmware versions. One byte of data returned is the version number, the bit 7 is 1, the bit 6 is 0, and the bits 5-0 are the version number. If the returned value is 0B7H, remove 1 of bit 7, and the version number will be 37H.

5.2. Command SET_BAUDRATE

This command is used to set the serial communication baud rate of CH375. When CH375 works in the serial communication mode, the default communication baud rate is 9600bps after reset. If MCU supports high communication speed, the serial communication baud rate can be dynamically regulated through this command. The command requires the input of two data, namely, baud rate frequency division coefficient and frequency division constant.

Usually, the serial port communication baud rate is set within 1mS. After completion, CH375 outputs the operation state at the newly set communication baud rate. Therefore, MCU shall adjust its own communication baud rate in time after sending the command.

The following table shows frequency division coefficients and frequency division constants corresponding to serial communication baud rates.

Frequency division coefficient	Frequency division constant	Serial communication baud rate (bps)	Error
02H	B2H	9600	0.16%
02H	D9H	19200	0.16%
03H	98H	57600	0.16%
03H	CCH	115200	0.16%
03H	F3H	460800	0.16%
07H	F3H	921600	0.16%
03H	C4H	100000	0%
03H	FAH	1000000	0%
03H	FDH	2000000	0%
02H	Constant	Calculation formula: 750000/(256- constant)	
03H	Constant	Calculation formula: 6000000/(256- constant)	

5.3. Command ENTER_SLEEP

This command enables CH375 to be in the low-power sleep suspended state (this function is not supported on some models of chips). After entering the low power state, the clock of CH375 stops vibrating, thus saving power. The low power state is not quitted until one of the following two situations is detected: first, the signal is detected on the USB bus; second, the signal is detected in the USB bus; second, MCU writes new commands to CH375 (commands without input data, such as GET_IC_VER or ABORT_NAK).

Typically, it takes several milliseconds for CH375 to exit the low-power state and return to normal operation. When fully restored to normal operation, CH375 will generate the event interrupt USB_INT_WAKE_UP.

5.4. Command RESET_ALL

This command enables CH375 to perform a hardware reset. Typically, hardware reset is completed within 40mS.

5.5. Command CHECK_EXIST

This command is used to test the working state to check whether CH375 is working properly. This command needs to input 1 data, which can be any data. If CH375 is working properly, the output data of CH375 will be the bitwise reverse of the input data. For example, if the input data is 57H, the output data will be A8H. In addition, CH375 normally reads the data 00H from its parallel port before receiving no command after its reset.

5.6. Command GET_MAX_LUN

This command is used to get the maximum logical unit number of the USB storage device. This command requires to input one data 38H and output the maximum logical unit number of the USB storage device. Some USB storage devices support the multilogic unit. The maximum logical unit plus 1 is the total number of logical units.

5.7. Command SET_DISK_LUN

This command is used to set the current logical unit number of the USB storage device. This command requires to input two data, respectively the data 34H and the current logical unit number. Some USB storage devices support multilogic unit. After CH375 initializes the USB storage device, it accesses 0# logical unit by default. If it is necessary to access other logical units, select the current logical unit number through this command.

5.8. Command SET_PKT_P_SEC

This command is used to set the total number of data packets per sector for the USB storage device. This command requires to input two data, respectively the data 39H and the total number of data packets per sector. After CH375 initializes the USB storage device, the size of each sector is 512 bytes by default. If the sector size is not 512 bytes through the command DISK_SIZE, the total number of data packets for each sector can be set through this command. Its value is the sector size divided by 64. For the 512-byte sector, the total number of data packets is 8; for a 2048-byte sector, the total number of data packets is 32.

5.9. Command SET_USB_MODE

This command is used to set USB working mode. This command needs to input 1 data, which is a mode code:

Switch to the disabled USB device mode (default mode after power-on or reset) when the mode code is 00H;

Switch to the enabled USB device mode and external firmware mode when the mode code is 01H;

Switch to the enabled USB device mode and built-in firmware mode when the mode code is 02H;

Switch to the disabled USB host mode when the mode code is 04H;
Switch to the enabled USB host mode when the mode code is 05H, not generate SOF package;
Switch to the enabled USB host mode when the mode code is 06H, automatically generate SOF package;
Switch to the enabled USB host mode when the mode code is 07H, resetting USB bus;

Please refer to CH372 Datasheet for USB device mode.

In USB host mode, "Not enabled" means that whether the USB device is connected is not automatically detected, so the external MCU shall detect; "Enabled" means that whether the USB device is connected is automatically detected. When the USB device is connected or disconnected, an interrupt will be generated to notify the external MCU. After switching to the mode code 06H, CH375 will automatically periodically generate a USB frame cycle SOF packet to be sent to a connected USB device. The mode code 07H is usually used to provide a USB bus reset state to a connected USB device, and the USB bus reset will not end until it is switched to other working mode. It is recommended to use mode 5 when there is no USB device. After the USB device is plugged, enter mode 7 first and then switch to mode 6.

Usually, the USB working mode is set within 20uS, and the operation status is output after completion.

5.10. Command TEST_CONNECT

This command is used to query the connection status of the current USB device in USB host mode. Typically, this command is completed within 2uS. After completion, USB_INT_CONNECT, USB_INT_DISCONNECT, or USB_INT_USB_READY is output. USB_INT_CONNECT indicates that the USB device is just connected or has been connected but has not been initialized. USB_INT_DISCONNECT indicates that the USB device has not been connected or has been disconnected. 0 indicates that the command is not completed and the status can be read later.

5.11. Command ABORT_NAK

This command is used to abort the retry of the current NAK. When CH375 works in the USB host mode, by default, it will keep retrying until it returns success or error when receiving the NAK status returned by the USB device. This command can force CH375 to terminate a retry in order to perform a new operation. In addition, the command SET_RETRY can be used to set whether NAK retry is disabled.

5.12. Command GET_STATUS

This command is used to get the interrupt status of CH375 and notify CH375 to cancel the interrupt request. After CH375 requests an interrupt from MCU, MCU gets the interrupt status through the command, analyzes the interrupt cause and processes.

Interrupt status byte	Classification of interrupt status
00H～0FH	Please refer to CH372 Datasheet for the interrupt status in USB device mode.
10H～1FH	Common interrupt statuses in USB host mode
20H～3FH	Operation failure status in USB host mode, used to analyze the cause of operation failure

The following table shows common interrupt statuses in USB host mode.

Status byte	Status name	Analysis and description of interrupt status
14H	USB_INT_SUCCESS	USB transaction or transmission operation is successful
15H	USB_INT_CONNECT	A USB device connection event is detected
16H	USB_INT_DISCONNECT	A USB device disconnection event is detected

17H	USB_INT_BUF_OVER	The data transmitted by USB is wrong or the buffer overflows due to too many data
1DH	USB_INT_DISK_READ	USB storage device performs read operation, and requests data read
1EH	USB_INT_DISK_WRITE	USB storage device performs write operation, and requests data write
1FH	USB_INT_DISK_ERR	USB storage device operation failed

The following table shows operation failure statuses in USB host mode, which are generally used to analyze the cause of operation failure.

Interrupt status byte	Name	Analysis and description of interrupt status
Bits 7-6	(Reserved bit)	Always 00
Bit 5	(Flag bit)	Always 1, indicating the operation failure status
Bit 4	IN transaction Sync flag	For IN transactions, if the bit is 0, data packets currently received are out of sync and the data may be invalid
Bits 3-0	Returned value of USB device that causes operation failure	1010 = Device returns NAK
		1110 = Device returns STALL
		XX00 = Device returns timeout. The device did not return
		Other value is PID returned by the device

5.13. Command RD_USB_DATA

This command is used to read the data block from the endpoint buffer of the current USB interrupt. In USB host mode, the endpoint buffer of the USB interrupt is the input buffer of the USB host endpoint. The output data firstly read is data block length, namely, the number of bytes of subsequent data streams. The effective value of the data block length is 0 to 64. If the length is not 0, MCU must read the subsequent data one by one from CH375.

5.14. Command WR_USB_DATA7

This command is used to write data blocks to the output buffer of the USB host endpoint or to the upload buffer of USB endpoint 2. The input data firstly written is data block length, namely, the number of bytes of subsequent data streams. The effective value of the data block length is 0 to 64. If the length is not 0, MCU must write the subsequent data one by one to CH375.

5.15. Command DISK_INIT

This command is used to initialize USB storage device. For the connected USB device, the command first resets the USB bus, then analyzes the descriptor of the USB device. If it is a supported USB storage device, the device will be automatically configured, and a connection to the USB storage device will be finally established. CH375 requests an interrupt from MCU after the command is executed. MCU can read the interrupt status as the operation status of the command. If the USB device has been disconnected, the operation status may be USB_INT_DISCONNECT. If the USB device is not recognized or the USB storage device is not supported, the operation status will be usually USB_INT_DISK_ERR or USB_INT_BUF_OVER. If the USB storage device is initialized successfully, the operation status will be USB_INT_SUCCESS.

5.16. Command DISK_SIZE

This command is used to get the physical capacity of the USB storage device. After successfully initializing

the USB storage device, this command gets the total capacity of the USB storage device. CH375 requests an interrupt from MCU after the command is executed. MCU can read the interrupt status as the operation status of the command. If the operation status is USB_INT_SUCCESS, the data can be gotten through the command RD_USB_DATA. The data is usually 8 bytes. The first 4 bytes constitute double-word data with high bytes in the front, which is the total number of sectors of USB storage device. The last 4 bytes constitute the double-word data with high bytes in the front, which is the number of bytes of each sector. The result of multiplying two data is the total capacity of USB storage device in bytes. If the sector size is not 512 bytes, the command SET_PKT_P_SEC shall be executed to set the total number of data packets per sector.

5.17. Command DISK_READ

This command is used to read data blocks from the USB storage device. Two parameters are required to read the data blocks in sectors: the start address and data length. The start address is the linear sector number LBA address expressed by 4 bytes, and the data length is the sector number expressed by 1 byte. This command requires to input 5 data, respectively the lowest byte of LBA address, the lower byte of LBA address, the higher byte of LBA address, the highest byte of LBA address and the number of sectors in turn. This command can read the data of 1-255 sectors at a time in the USB storage device with the capacity up to 2000GB. This command shall be used in conjunction with the command DISK_RD_GO below.

5.18. Command DISK_RD_GO

This command enables CH375 to continue to execute the read operation of USB storage device. After MCU sends the command DISK_READ, every time CH375 reads 64 bytes of data from USB storage device, CH375 will request interrupt, and MCU will get the interrupt status USB_INT_DISK_READ, so MCU shall send the command RD_USB_DATA to take 64 bytes of data, and then send the command DISK_RD_GO to make CH375 continue to read. CH375 reads 64 bytes of data from the USB storage device again and requests an interrupt again, and MCU takes the data again and make CH375 continue to read again. Until all data is fully read, CH375 will request an interrupt for the last time. MCU gets the interrupt status as the status of the whole read operation. If the operation is successful, the status will be USB_INT_SUCCESS; otherwise, it may be USB_INT_DISK_ERR.

Even if the command DISK_READ sent by MCU only reads 1 sector, MCU will normally receives (total number of data packets per sector + 1) interrupts (9 interrupts in total when the sector size is 512 bytes). The front interrupts require MCU to take data, and the last interrupt is to return the final operation status. If 4 sectors are read, MCU will normally receive (4 * total number of data packets per sector + 1) interrupts, the front interrupts require MCU to take data. If the read operation fails midway, MCU may receive the status USB_INT_DISK_ERR in advance, so as to end the read operation in advance.

5.19. Command DISK_WRITE

This command is used to write data blocks to the USB storage device. Two parameters are required to write the data blocks in sectors: the start address and data length. The start address is the linear sector number LBA address expressed by 4 bytes, and the data length is the sector number expressed by 1 byte. This command requires to input 5 data, respectively the lowest byte of LBA address, the lower byte of LBA address, the higher byte of LBA address, the highest byte of LBA address and the number of sectors in turn. This command can write the data of 1-255 sectors at a time in the USB storage device with the capacity up to 2000GB. This command shall be used in conjunction with the command DISK_WR_GO below.

5.20. Command DISK_WR_GO

This command enables CH375 to continue to execute the write operation of USB storage device. After MCU sends the command DISK_WRITE, CH375 will request interrupt quickly, and MCU will get the interrupt status USB_INT_DISK_WRITE, so MCU shall send the command WR_USB_DATA7 to provide 64-byte

data, and then send the command DISK_WR_GO to make CH375 continue to write. Whenever CH375 writes 64 bytes of data to the USB storage device, it will request an interrupt again, and MCU will provide the data again and make CH375 continue to write again. Until all data is fully written, CH375 will request an interrupt for the last time. MCU gets the interrupt status as the status of the whole write operation. If the operation is successful, the status will be USB_INT_SUCCESS; otherwise, it may be USB_INT_DISK_ERR.

Even if the command DISK_WRITE sent by MCU only writes 1 sector, MCU will normally receives (total number of data packets per sector + 1) interrupts (9 interrupts in total when the sector size is 512 bytes). The front interrupts require MCU to provide data, and the last interrupt is to return the final operation status. If 4 sectors are written, MCU will normally receive (4 * total number of data packets per sector + 1) interrupts, the front interrupts require MCU to provide data. If the write operation fails midway, MCU may receive the status USB_INT_DISK_ERR in advance, so as to end the write operation in advance.

5.21. Command DISK_INQUIRY

This command is used to inquire the features of USB storage device. CH375 requests an interrupt from MCU after the command is executed. MCU can read the interrupt status as the operation status of the command. If the interrupt status is USB_INT_SUCCESS, the data can be gotten through the command RD_USB_DATA. The data is usually 36 bytes, including the features of USB storage device and the identification information of manufacturer and product. This command is generally not needed unless a new logical unit is analyzed.

5.22. Command DISK_READY

This command is used to check whether the USB storage device is ready. CH375 requests an interrupt from MCU after the command is executed. MCU can read the interrupt status as the operation status of the command. If the operation status is USB_INT_SUCCESS, it will indicate that the USB storage device has been ready.

5.23. Command DISK_R_SENSE

This command is used to check the error of the USB storage device. CH375 requests an interrupt from MCU after the command is executed. MCU can read the interrupt status as the operation status of the command. The operation status is normally USB_INT_SUCCESS, and the error can be analyzed after the data is gotten through the command RD_USB_DATA.

6. Functional Specification

6.1. General Description

CH375 can work in USB-HOST mode or USB device mode.

The USB device mode of CH375 is fully compatible with CH372. For related information, refer to CH372 Datasheet.

The USB host mode of CH375 supports both parallel and serial interfaces. In USB host mode, CH375 supports commonly used USB full-speed devices, and the external MCU needs to write the firmware program to communicate with USB devices according to the corresponding USB protocol. However, for USB storage devices, CH375 has related protocols. Normally, the external MCU does not need to write the firmware program to communicate directly.

6.2. Hardware of Local End

CH375 provides a universal passive parallel interface and a point-to-point serial interface at the local end.

During the reset of the CH375, TXD pin is used to select the communication interface. If CH375 detects that

TXD pin is at low level during reset, the parallel interface will be enabled; otherwise, the serial interface will be enabled. If the serial interface is enabled, TXD pin will be used for serial data output after reset, and CH375 can only work in USB host mode.

6.2.1. Parallel Interface

The parallel port signal line includes: 8-bit bidirectional data buses D7-D0, read strobe input pin RD#, write strobe input pin WR#, chip selection input pin CS#, interrupt input pin INT# and address input pin A0. Through the passive parallel interface, CH375 can be easily hooked to the system bus of 8-bit DSP or MCU, and can coexist with a number of peripheral devices.

CS# of CH375 is driven by the address decoding circuit, and can be used for device selection when MCU has multiple peripheral devices. The interrupt request of INT# output is active at low level and can be connected to the interrupt input pin or ordinary I/O pin of MCU. MCU can get the interrupt request in interrupt mode or query mode.

For MCU similar to the Intel parallel port timing sequence, RD# and WR# pins of CH375 can be connected to the read strobe output pin and write strobe output pin of MCU respectively. For MCU similar to Motorola parallel port time sequence, the RD# pin of the CH375 shall be connected to the low level, and the WR# pin shall be connected to the reading and writing direction output pin R/-W of MCU.

CH375 occupies two address bits. When A0 pin is at high level, write a new command, or read the interrupt flag; when A0 pin is at low level, select the data port to read and write the data.

The following table is the truth table of the parallel port I/O operation (X in the table means that this bit is not concerned, and Z means that three states of CH375 are disabled).

CS#	WR#	RD#	A0	D7-D0	Actual operation on CH375
1	X	X	X	X/Z	CH375 is not selected, and no any operation is made
0	1	1	X	X/Z	Although selected, no any operation is made
0	0	1/X	1	Input	Write a command code to the command port of CH375
0	0	1/X	0	Input	Write data to the data port of CH375
0	1	0	0	Output	Read data from the data port of CH375
0	1	0	1	Output	Read the interrupt flag from the command port of CH375B/C, and the bit 7 is equivalent to INT# pin

6.2.2. Serial Interface

The serial interface can only be used in USB host mode. The USB device mode of CH375 does not support the serial interface.

The signal line of the serial interface includes serial data input pin RXD, serial data output pin TXD and interrupt output pin INT#. CH375 can be connected point-to-point to DSP and MCU through a serial interface by using less connecting wires over a longer distance.

RXD and TXD of CH375 can be connected to the serial data output pin and serial data input pin of MCU respectively. The interrupt request output by INT# is active at low level and used to notify MCU.

The serial data format of CH375 is 1 start bit, 9 data bits, and 1 stop bit, in which the first 8 data bits are 1 byte of data and the last data bit is a command flag bit. When the bit 9 is 0, the data of the first 8 bits are written to CH375; when the bit 9 is 1, the first 8 bits are written to CH375 as command codes. The serial communication baud rate of CH375 is 9600bps by default. MCU can select the appropriate communication baud rate through the command SET_BAUDRATE at any time.

6.2.3. Others

ACT# pin of CH375 is used for the status indication. After the USB device is not configured or is unconfigured in USB device mode of internal firmware, this pin will output a high level; when the USB device is configured, this pin will output a low level. In the USB host mode, when the USB device is disconnected, the pin outputs a high level; when the USB device is connected, the pin outputs a low level. ACT# pin of CH375 can be externally connected to LED with a current limiting resistor connected in series to indicate the relevant status.

The UD+ and UD- pins of CH375 are USB signal lines, which shall be directly connected to the USB bus when working in USB device mode; yet they can be directly connected to USB device when working in USB host mode. If a fuse resistor or inductor or ESD protection device is connected in series for chip safety, the AC and DC equivalent series resistors shall be within 5Ω.

CH375 has a built-in power on reset circuit. Generally, no external reset is required. RSTI pin is used to input an asynchronous reset signal from the outside; when RSTI pin is at high level, CH375 will be reset; when RSTI pin recovers to a low level, CH375 will continuously delay reset for about 20-35mS, and then enter the normal working status. In order to reliably reset and reduce external interference during the power-on period, a capacitor with a capacity of about 0.1uF can be connected across the RSTI pin and VCC. RST pin and RST# pin are reset status output pins, which are respectively active at high level and active at low level; RST pin and RST# pin output high level and low level respectively when CH375 power supply is switched on or externally forced to be reset and during reset delay. After CH375 reset is completed, RST pin and RST# pin respectively restore to low and high levels. RST and RST# pins can be used to provide power on reset signals to the external MCU.

When CH375B chip works normally, 12MHz clock signal shall be provided for it externally. Generally, the clock signal is generated by the built-in inverter of CH375 through the crystal stable frequency oscillator. The peripheral circuit is only required to be connected with a crystal with a nominal frequency of 12MHz between XI and XO pins, and connected with a high frequency oscillating capacitor to the ground for XI and XO pins respectively. If the 12MHz clock signal is inputted directly from the outside, it shall be inputted from the XI pin, and the XO pin is suspended.

CH375C chip supports two modes: external clock and built-in clock. For the external clock mode, refer to the aforesaid external 12MHz crystal and capacitor of CH375B; in the built-in clock mode, the XI pin shall be connected to GND and the XO pin shall be suspended to omit the external crystal and the oscillation capacitor.

CH375B and CH375C chips support supply voltage of 3.3V or 5V. When a 5V operating voltage is used, VCC pin of CH375 will input an external 5V power supply, and V3 pin shall be connected to an external power decoupling capacitor with a capacity of 0.01uF to 0.1uF. When 3.3V operating voltage is used, the V3 pin of CH375 shall be connected to the VCC pin, and an external 3.3V power supply shall be inputted at the same time, and the operating voltage of other circuits connected to CH375 shall not exceed 3.3V.

6.3. Internal Structure

CH375 integrates PLL frequency multiplier, master-slave USB interface SIE, data buffer, passive parallel interface, asynchronous serial interface, command interpreter, protocol processor for control transmission, general firmware program, etc.

PLL frequency multiplier is used to multiply 12MHz external input clock frequency to 48MHz as the USB interface SIE clock.

Master-slave USB interface SIE is an SIE with USB host mode and USB device mode, which is used for completion of physical USB data receiving and sending, automatic processing of bit tracking and synchronization, NRZI encoding and decoding, bit stuffing, conversion between parallel data and serial data,

CRC data check, transaction handshake, error retry and USB bus status detection, etc.

The data buffer is used to buffer data sent and received by USB interface SIE.

The passive parallel interface is used to exchange data with the external DSP/MCU.

The asynchronous serial interface is used to exchange data with the external DSP/MCU instead of the passive parallel interface.

The command interpreter is used to analyze and execute various commands submitted by DSP/MCU.

The control transmission protocol processor is used to automatically process multiple stages of commonly used control transmissions and simplify external firmware programming.

The general-purpose firmware programs include two sets: one for USB device mode, automatically processing various standard transactions of USB default endpoint 0; the other for USB host mode, automatically processing the dedicated communication protocol of Mass-Storage devices.

There are 7 physical endpoints inside CH375:

Endpoint 0 is the default endpoint and supports upload and download. The size of upload and download buffers is respectively 8 bytes.

Endpoint 1 includes the upload endpoint and the download endpoint. The size of upload and download buffers is respectively 8 bytes. The number of the upload endpoint is 81H and the number of the download endpoint is 01H;

Endpoint 2 includes the upload endpoint and the download endpoint. The size of upload and download buffers is respectively 64 bytes. The number of the upload endpoint is 82H and the number of the download endpoint is 02H.

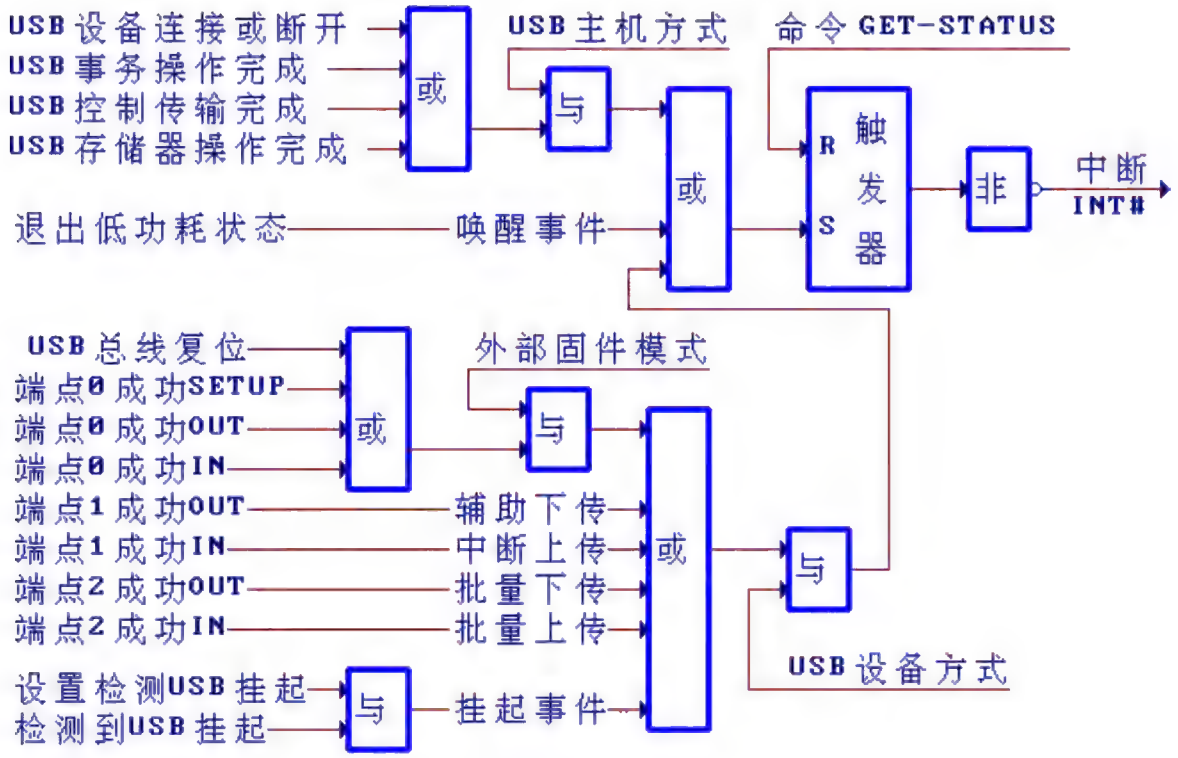
The host endpoint includes the output endpoint and the input endpoint. The size of output and input buffers is respectively 64 bytes, and the host endpoint and the endpoint 2 share the same set of buffers. The output buffer of the host endpoint is the upload buffer of the endpoint 2, and the input buffer of the host endpoint is the download buffer of the endpoint 2.

Endpoint0, endpoint1, and endpoint2 of CH375 are only used in USB device mode, and only host endpoints are used in USB host mode.

In USB host mode, CH375 supports a variety of commonly used USB full-speed devices. The endpoint number of USB devices can be 0 to 15, up to 31 endpoints can be supported in both directions, and the package length of USB devices can be 0-64 bytes.

The built-in firmware can process the communication protocol of mass-Storage devices, and requires USB storage devices to support Bulk-Only transport protocol, SCSI, UFI, RBC or equivalent command set, and the maximum packet length of the data endpoint is 64 bytes, but the maximum packet length of the default endpoint 0 can be 8, 16, 32 or 64 bytes. If the USB storage device does not meet the above requirements, the external MCU is required to process the relevant communication protocols through the control transmission and the command ISSUE_TOKEN or ISSUE_TKN_X.

The figure below shows the interrupt logic inside CH375.



6.4. MCU Software of Local End

MCU reads and writes CH375 through an 8-bit parallel port. All operations are composed of a command code, several input data and several output data. Some commands do not need input data, and some commands do not have output data. The command operation steps are as follows:

① Write the command code to the command port when A0 is 1;

② If the command has input data, write the input data in sequence when A0 is 0, one byte at a time;

③ If the command has output data, read the output data in sequence when A0 is 0, one byte at a time;

④ The command is completed. Pause or go to ① to continue to execute the next command.

CH375 is specially used to process USB communication. When detecting the status change of USB bus or executing the command, CH375 will notify MCU to process in an interrupt mode.

7. Parameters

7.1. Absolute Maximum Value

Critical value or exceeding the absolute maximum value may cause the chip to work abnormally or even be damaged.

Name	Parameter description		Min.	Max.	Unit
TA	Ambient temperature during operation	CH375B or CH375C chip, VCC=5V, external clock	-40	85	℃
		CH375B or CH375C chip, VCC=3.3V, external clock	-40	85	
		CH375C chip, VCC=5V or 3.3V, internal clock	-20	70	
TS	Ambient temperature during storage		-55	100	℃
VCC	Supply voltage (VCC connects to power, GND to ground)		-0.5	6.0	V
VIO	Voltage on the input or output pins		-0.5	VCC+0.5	V

7.2. Electrical Parameters

Test Conditions: TA=25°C, VCC=5V, Excluding the Pins Connected to the USB Bus
(If the supply voltage is 3.3V, all current parameters in the table need to be multiplied by a factor of 40%)

Name	Parameter description		Min.	Typ.	Max.	Unit
VCC	Supply voltage	CH375B/C chip, V3 not connected to VCC	4.2	5	5.4	V
		CH375B/C chip, V3 is connected to VCC	3.1	3.3	3.6	
ICC	Total supply current during operation	CH375B chip, VCC=5V		11	30	mA
		CH375C chip, VCC=5V		7	25	
		CH375B/C chip, VCC=3.3V		5	15	
ISLP	Supply current at the low power status I/O pin suspended/ internal pull-up	VCC=5V		0.15	0.25	mA
		VCC=3.3V		0.05	0.15	
VIL	Low level input voltage		-0.5		0.7	V
VIH	High level input voltage		2.0		VCC+0.5	V
VOL	Low level output voltage (4mA draw current)				0.5	V
VOH	High level output voltage (4mA output current)		VCC-0.5			V
IUP	Input current at the input terminal of built-in pull-up resistor		40	80	160	uA
IDN	Input current at the input terminal of built-in pull-down resistor		-80	-140	-240	uA
VR	Voltage threshold of power-on reset		2.1	2.6	3.0	V

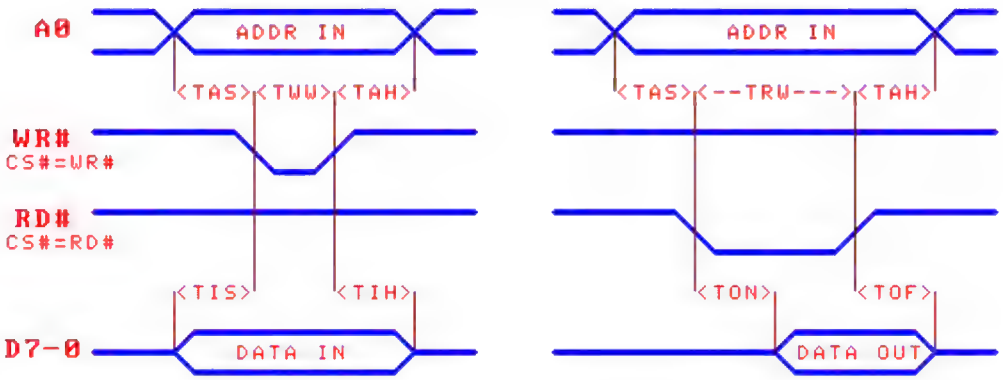
Note: Low level draw current of ACT# pin is 4mA, and the high level output current is 200uA.
During CH375 reset, INT# and TXD pins only provide the high level output current of 80uA.

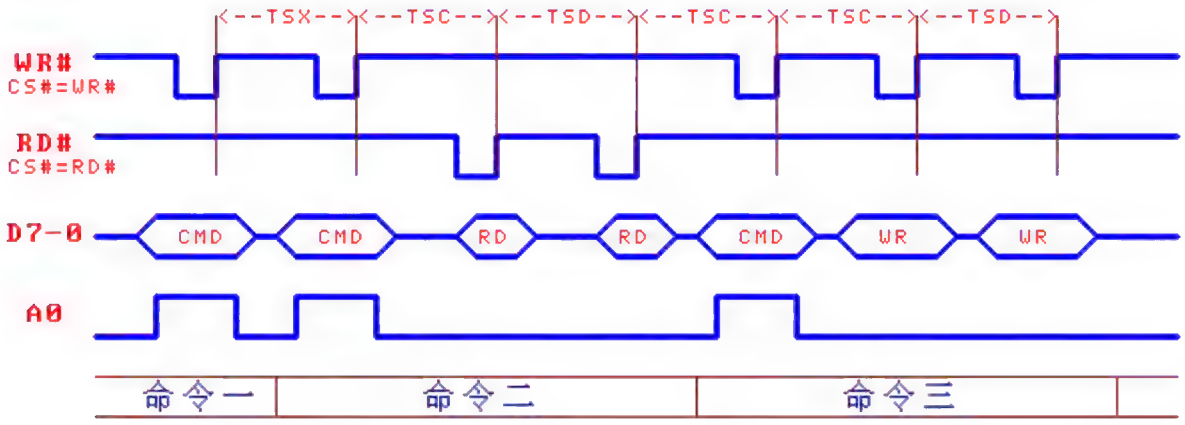
7.3. Timing Parameters

Test Conditions: TA=25°C, VCC=5V, refer to the attached figure.
(RD means that RD# signal is valid and CS# signal is valid; RD#=CS#=0 performs read operation)
(WR means WR# signal is valid and CS# signal is valid, WR#=CS#=0 performing a writing operation)

Name	Parameter description	Min.	Typ.	Max.	Unit
FCLK	Input clock frequency of XI pin in USB host mode	11.995	12.00	12.005	MHz
FCLKI	Internal frequency in CH375C built-in clock mode	11.82	12.00	12.18	MHz
TPR	Internal power-on reset time	10	30	40	mS
TRI	Effective signal width of external reset input	100			nS
TRD	Reset delay after external reset input	12	30	40	mS
TE1	Execution time of command RESET_ALL	0.1	30	40	mS
TE2	Execution time of command SET_USB_MODE		10	20	uS
TE3	Execution time of command SET_ENDP?		2	4	uS

TE4	Execution time of command SET_BAUDRATE	100		1000	uS
TE0	Execution time of other commands		1.5	2	uS
TSX	Interval time between CH375B/C command codes	1.5			uS
TSC	Interval time between CH375B/C command code and data	1.5			uS
TSD	Interval time between CH375B/C data and data	0.6			uS
TWW	Width of effective strobe writing pulse WR	60		10000	nS
TRW	Width of effective strobe reading pulse RD	60		10000	nS
TAS	Address input setup time before RD or WR	5			nS
TAH	Address input hold time after RD or WR	5			nS
TIS	Data input setup time before write strobe WR	0			nS
TIH	Data input hold time after strobe writing WR	5			nS
TON	Data valid after Read LOW	0		30	nS
TOF	Data hold after Read HIGH	0		20	nS
TINT	Receive the command GET_STATUS until INT# pin cancel the interrupt		1.5	3	uS
TWAK	Wake-up time when exiting from low-power state	2	6	10	mS





8. Application

8.1. Parallel Port Mode (Figure below)

This is a connection circuit between CH375 and the ordinary MCS-51 MCU. TXD pin of CH375 is grounded through pull-down resistor with resistance of about 1KΩ or directly grounded, so that CH375 works in parallel mode.

The USB bus includes a pair of 5V power lines and a pair of data signal lines. Generally, the +5V power line is red, the ground line is black, the D+ signal line is green, and the D- signal line is white. USB socket P1 can be connected to the USB device directly. If necessary, a high-speed electronic switches with current limiting function can be connected to +5V power line provided to USB devices in series. USB supply voltage must be 5V.

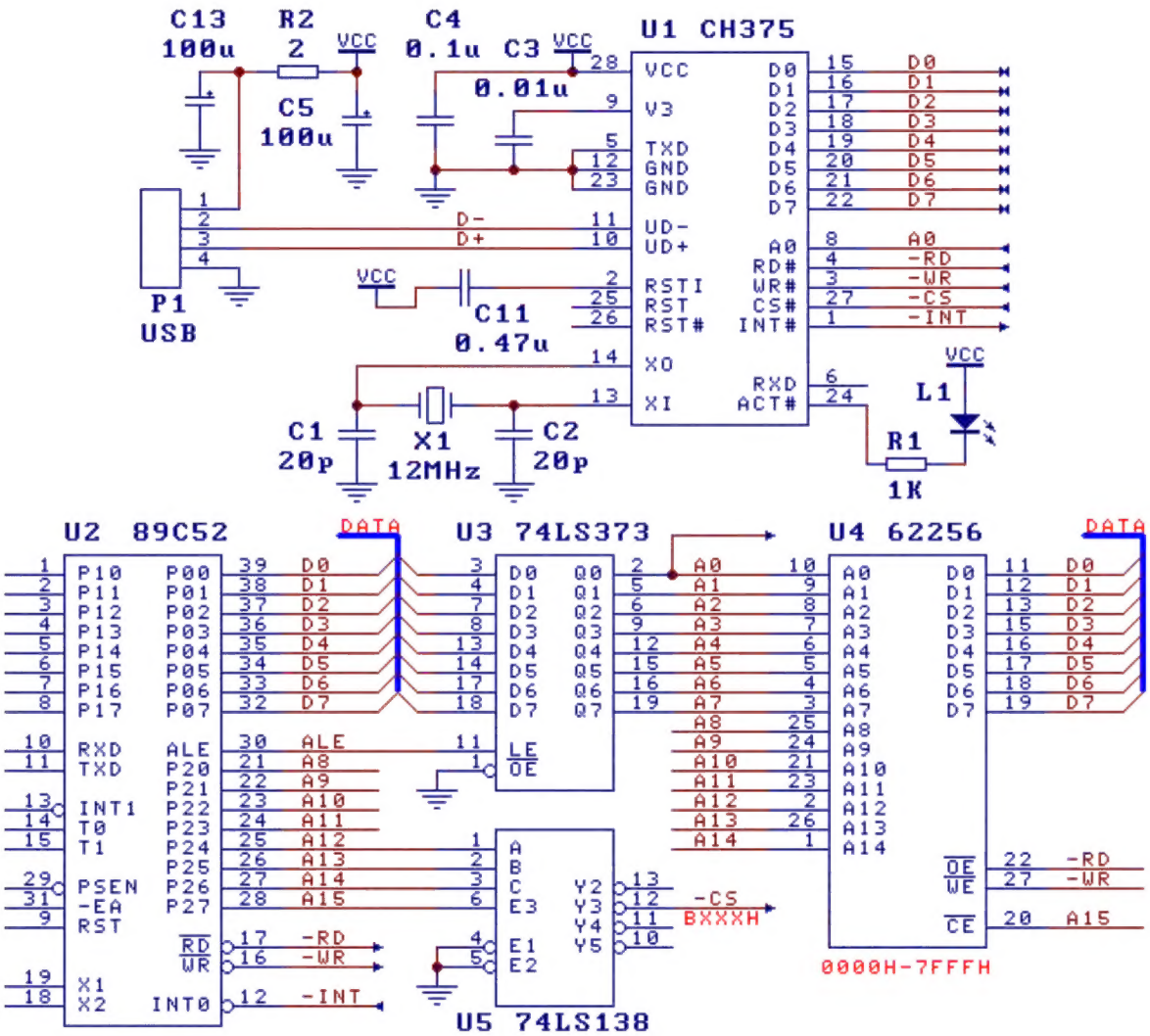
Capacitor C3 is used for decoupling the internal power node of CH375. C3 is a monolithic or high-frequency ceramic capacitor with a capacity of 0.01μF to 0.1μF. Capacitors C4 and C5 are used for decoupling the external power supply, and C4 is a monolithic or high-frequency ceramic capacitor with a capacity of 0.1μF. Crystal X1, capacitors C1 and C2 are used in the clock oscillation circuit of CH375. The USB-HOST mode requires an accurate clock frequency. The frequency of crystal X1 is 12MHz±0.4‰, and C1 and C2 are monolithic or high-frequency ceramic capacitors with a capacity of 15pF～30pF.

If CH375B chip is replaced with CH375C with a built-in clock for the finished board, remove X1 and C1, and replace C2 with a 0Ω resistor.

To ensure that CH375 is reset reliably, the rise time of the supply voltage from 0V to 5V shall be less than 100mS. CH375 will not be reliably reset if the power supply is switched on slowly and does not discharge in time after the power supply is switched off. The capacitor C11 with a capacity of 0.1μF or 0.47μF can be connected across the RSTI pin and VCC to extend the reset time.

If the supply voltage of CH375 is 3.3V, V3 pin shall be short-circuited with VCC pin to jointly input 3.3V voltage, and the capacitor C3 can be eliminated.

It shall be noticed that the decoupling capacitors C3 and C4 shall be as close as possible to the connected pins of CH375 when the printed circuit board PCB is designed; the D+ and D- signal lines shall be close to parallel wiring, and ground wire or covered copper shall be provided on both sides to reduce the external signal interference; the length of the signal lines related to the XI and XO pins shall be shortened as far as possible to reduce the interference. The ground wire or covered copper shall surround the relevant components.



CH375 has a universal passive parallel interface and can be directly connected to a variety of DSP, MCU, etc. In the typical application circuit of MCS-51 series MCU, CH375 can be directly hooked to the system bus of MCU U2 through D7-D0, -RD, -WR, -CS and A0 of the 8-bit passive parallel interface.

If MCS-51 MCU does not use U3 to latch the addresses A7-A0, the address line A0 of CH375 can be driven through P20 and other pins of U2, and the port address in MCU program shall be modified accordingly. U4 is used for simple address decoding to generate the required chip selection signal. In the figure, the chip selection address range of CH375 is B000H-BFFFH. In fact, CH375 only needs to occupy two addresses: address BXX1H for writing commands and address BXX0H for reading and writing data.

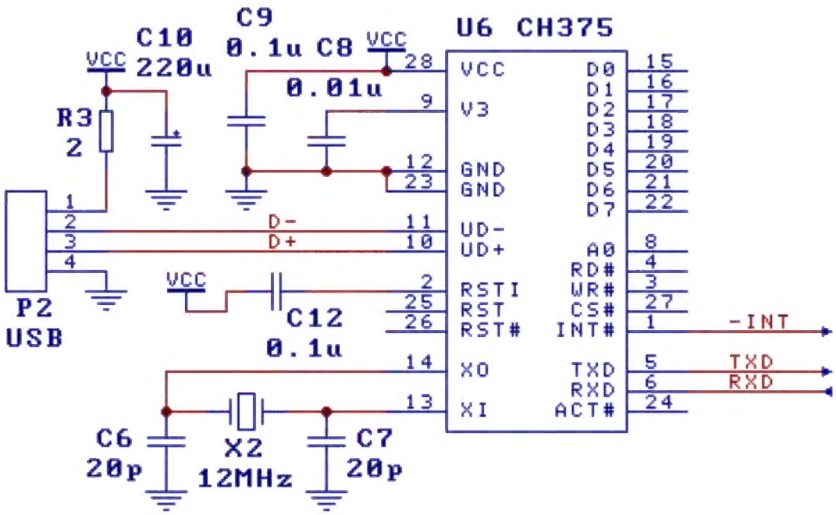
8.2. Serial Mode (Figure below)

If TXD pin of CH375 is suspended or not grounded through a pull-down resistor, CH375 will work in serial port mode. In the serial port mode, CH375 only needs to be connected to 3 signal lines with DSP/MCU: TXD pin, RXD pin and INT# pin. Other pins can be suspended. Other peripheral circuits are basically the same as that in the parallel port mode except for fewer connections.

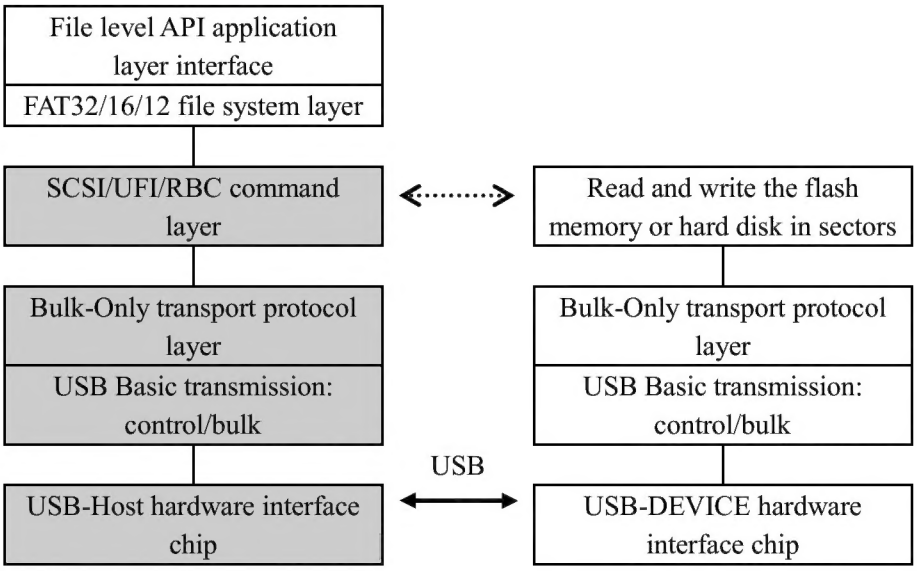
In addition, if the communication baud rate of CH375 serial port is required to be dynamically modified, it will be suggested that the I/O pin of MCU control RSTI pin of CH375, so as to reset CH375 to return to the default communication baud rate when necessary. Because RSTI pin has a pull-down resistor, a pull-up resistor with resistance of several KΩ may be required to be added when the quasi-bidirectional I/O pin of

MCS51 MCU drives it.

As INT# pins and TXD pin can only provide weak high level output current during CH375 reset, when a long distance connection is conducted, in order to avoid INT# or TXD interference caused by MCU misoperation during CH375 reset, a pull-up resistor with resistance of 2KΩ-5KΩ can be added on INT# pin or TXD pin to maintain a stable high level. After CH375 reset is completed, INT# and TXD pins will be able to provide either a high level output current of 4mA or a low level draw current of 4mA.



8.3. MCU Read/Write files in USB Flash Disk (File-Level Interface of USB Storage Device)



In general, it is necessary to realize the four levels on the left in the above figure for MCU or embedded system to process the file system of the USB flash disk. The internal structure hierarchy of the USB storage device is shown on the right. As CH375 not only is a general-purpose USB-HOST hardware interface chip, but also has related firmware program, containing 3 levels (parts marked in gray) on the left of the above figure, the actual MCU program only needs to process FAT file system layer, and even this layer can be realized by USB flash disk file level subroutine library of CH375.

If the file system is not required to be processed, that is, the top layer on the left of the above figure is not processed, CH375 will directly provide the data block read-write interface by taking 512-bytes or 2K-byte physical sector as the basic read and write unit, thus simplifying the USB storage device to an external data memory. MCU can freely read and write data in USB storage devices and also freely define the data

structure.

As the computer organizes the USB storage device into a file system, MCU can also organize the USB storage device into a file system, that is, process the top layer on the left of the figure above, in order to facilitate the exchange of data between MCU and the USB mobile storage device.

CH375 provides file level interfaces of USB storage devices in C language subroutine library. API of these application layer interface contains common file level operations, which can be transplanted and embedded into various common MCU programs.

USB flash disk file level subroutine library of CH375 has the following features: it supports commonly used FAT12, FAT16 and FAT32 file systems, has the disk capacity of more than 100GB, supports multi-level subdirectories, supports 8.3 format uppercase and Chinese filenames, supports lowercase or long filenames, supports file open, create, delete, read and write, and search, etc.

The file level subroutine library of CH375 requires at least 600 bytes of RAM as a buffer. Taking the ordinary MCS-51 MCU as an example, all subroutines of the file system have 4KB to 8KB code, and need about 80 bytes of internal RAM and at least 512 bytes of external RAM as buffers. For more information about USB flash disk file level subroutine library, refer to the instructions on CH375 evaluation board.

All APIs in the file level subroutine library have operation status returned after call, but response data not necessarily returned. Please refer to CH375HF?.H for the description of API parameters. The main subroutines are as follows:

Initialize CH375: CH375Init
Inquire whether the USB flash disk is ready: CH375DiskReady
Inquire the capacity of USB flash disk: CH375DiskSize
Inquire the information of USB flash disk: CH375DiskQuery
Open file: CH375FileOpen
Enumeration or search file: CH375FileEnumer
Close file: CH375FileClose
Create file: CH375FileCreate
Delete file: CH375FileErase
Read data from the file in sectors: CH375FileReadX
Write data to the file in sectors: CH375FileWriteX
Move the file pointer in sectors: CH375FileLocate
Inquire file attributes (attribute/date/time/length): CH375FileQuery
Set file attributes (attribute/date/time/length): CH375FileModify
Read data from the file in bytes: CH375ByteRead
Write data to the file in bytes: CH375ByteWrite
Move the file pointer in bytes: CH375ByteLocate